%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 优化算法开始

NPar = 2; % 优化参数2个

VarLow =[0 -2]; % 加速度下限

VarHigh =[2 0]; % 加速度上限

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*参数

v\_base=30; %cruise speed

% ax\_1=0.2; pulse

% ax\_2=-1; glide

ess\_init\_soc=0.9;

PopSize = 5; %粒子数量

MaxIterations = 10;%迭代次数

KeepPercent = 20/100; %基因不变比例

CrossPercent = 80/100; %基因变异比例

MutatPercent = 1 - KeepPercent - CrossPercent;

SelectionMode = 1;

KeepNum = round(KeepPercent \* PopSize);

CrossNum = round(CrossPercent \* PopSize);

if mod(CrossNum,2) ~= 0

CrossNum = CrossNum + 1;

end

MutatNum = PopSize - KeepNum - CrossNum;

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*初始化种群

AA=rand(PopSize,NPar);

AB=VarHigh - VarLow;

ABC=[];

for i=1:NPar

C=AA(:,i)\*AB(:,i)+VarLow(1,i);

ABC=[ABC C];

end

Pop = ABC;

clear AA AB ABC C;

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*初始化种群适应度计算

for i=1:PopSize

ax\_1=Pop(i,1);

ax\_2=Pop(i,2);

sim('ACC\_CC\_model\_best\_2017bV2.mdl' );

error(i)= soc(1)-soc(end); %SOC消耗越少越好

end

[Cost Indx] = sort(error);

Pop = Pop(Indx,:);

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*主循环

MinMat =min(Cost);

for Iter = 1:MaxIterations

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*selec keep

Pop(1:KeepNum,:) = PSO\_Fcn(Pop(1:KeepNum,:));

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*交叉

SlectedIndexes = SelectParents\_Fcn(Cost,CrossNum,SelectionMode);

CrossPop = [];

for ii = 1:2:CrossNum

Par1Index = SlectedIndexes(ii);

Par2Index = SlectedIndexes(ii+1);

Par1 = Pop(Par1Index,:);

Par2 = Pop(Par2Index,:);

[Off1 , Off2] = CrossOver\_fcn(Par1,Par2);

CrossPop = [CrossPop ; Off1 ; Off2];

end

Pop(KeepNum+1:KeepNum+CrossNum,:) = CrossPop;

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*基因突变

AA=rand(MutatNum,NPar);

AB=VarHigh - VarLow;

ABC=[];

for i=1:NPar

C=AA(:,i)\*AB(:,i)+VarLow(1,i);

ABC=[ABC C];

end

Pop(KeepNum+CrossNum+1 : end , :)= ABC;

clear AA AB ABC;

[m,n] = size(Pop);

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*新的种群

for i=1:n

jj=1;

while jj<=m

if Pop(jj,i)>VarHigh(1,i)

Pop(jj,i)=VarHigh(1,i);

end

if Pop(jj,i)<VarLow(1,i)

Pop(jj,i)=VarLow(1,i);

end

jj=jj+1;

end

end

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*适应度计算start

for i=1:PopSize

ax\_1=Pop(i,1);

ax\_2=Pop(i,2);

sim('ACC\_CC\_model\_best\_2017bV2.mdl' );

error(i)= soc(1)-soc(end); %SOC消耗越少越好

end

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*%适应度计算end

[Cost Indx] = sort(error);

Pop = Pop(Indx,:);

MinMat = [MinMat min(Cost)];

end

%% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BestSolution = Pop(1,:) %最优变量

BestCost = Cost(1) %最优适应度值

plot(MinMat);xlabel ??;ylabel ????

save youhuadata.mat